

# Solving underdetermined systems with error-correcting codes\*

Ted Hurley<sup>†</sup>

## Abstract

In an underdetermined system of equations  $Ax = y$ , where  $A$  is an  $m \times n$  matrix, only  $u$  of the entries of  $y$  with  $u < m$  are known. Thus  $E_j w$ , called ‘measurements’, are known for certain  $j \in J \subset \{0, 1, \dots, m-1\}$  where  $\{E_i, i = 0, 1, \dots, m-1\}$  are the rows of  $A$  and  $|J| = u$ . It is required, if possible, to solve the system uniquely when  $x$  has at most  $t$  non-zero entries with  $u \geq 2t$ .

Here such systems are considered from an error-correcting coding point of view. The unknown  $x$  can be shown to be the error vector of a code subject to certain conditions on the rows of the matrix  $A$ . This reduces the problem to finding a suitable decoding algorithm which then finds  $x$ .

Decoding workable algorithms are shown to exist, from which the unknown  $x$  may be determined, in cases where the known  $2t$  values are evenly spaced (that is, when the elements of  $J$  are in arithmetic progression) for classes of matrices satisfying certain row properties. These cases include Fourier  $n \times n$  matrices where the arithmetic difference  $k$  satisfies  $\gcd(n, k) = 1$ , and classes of Vandermonde matrices  $V(x_1, x_2, \dots, x_n)$  (with  $x_i \neq 0$ ) with arithmetic difference  $k$  where the ratios  $x_i/x_j$  for  $i \neq j$  are not  $k^{\text{th}}$  roots of unity. The decoding algorithm has complexity  $O(nt)$  and in some cases, including the Fourier matrix cases, the complexity is  $O(t^2)$ .

Matrices which have the property that the determinant of any square submatrix is non-zero are of particular interest. Randomly choosing rows of such matrices can then give  $t$  error-correcting pairs to generate a ‘measuring’ code  $C^\perp = \{E_j | j \in J\}$  with a decoding algorithm which finds  $x$ .

This has applications to signal processing and compressed sensing.

## 1 Introduction

Underdetermined systems  $Aw = y$  are considered where  $A$  is an  $m \times n$  matrix,  $w$  an  $n \times 1$  unknown vector and  $u$  entries of  $y$  are known with  $u < m$ . It is given that  $w$  has at most  $t$  non-zero entries and that  $u \geq 2t$ . Thus the vector  $w = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$  is known to have at most  $t$  non-zero entries but the positions and the values of these non-zero entries are unknown.

Let the rows of  $A$  be denoted by  $\{E_0, E_1, \dots, E_{n-1}\}$ . Hence  $E_j w$  are taken or known for  $j \in J = \{j_1, j_2, \dots, j_u\}$  where  $u \geq 2t$ , and the problem is to determine  $w$ , if possible, from the ‘measurements’  $\{E_j w, j \in J\}$ . These measurements are sometimes referred to as ‘samples of  $w$ ’.

This has applications to signal processing and compressed sensing for which there is a huge and extensive literature. A signal may be measured or sampled by rows of a matrix. The work by Candès, Romberg and Tao, [3], is a basic reference for recent treatments of compressed sensing.

Here a linear algebra approach is taken based on error-correcting codes. It is shown that when the  $\{E_j | j \in J\}$  generate a code  $C^\perp$  such that the distance  $d(C)$  of the dual code  $C$  satisfies  $d(C) \geq 2t + 1$ , where  $t$  is the maximum number of non-zero elements of  $w$ , then  $w$  can be obtained by decoding. The problem then is to find a suitable decoding algorithm which is efficient and stable.

A general algorithm, which is a decoding process in disguise, for cases where there exist *error-correcting pairs* for  $C$  (see section 2.4 below for definition) is developed in Section 4. In certain cases when the measurements are evenly spaced, and with additional properties on the rows of  $A$ , error-correcting pairs are explicitly shown to exist. In these cases an explicit decoding algorithms are given in Algorithm 4.1 and in Algorithm 4.2. Evenly spaced here means that the elements in  $J$  are in arithmetic sequence.

---

\*MSC Classification: 94A99, 15B99

Keywords: Underdetermined system, Codes, Signal processing, Compressed sensing

<sup>†</sup>National University of Ireland Galway, email: Ted.Hurley@NUIGalway.ie

When the arithmetic difference  $k$  (in  $J$ ) satisfies  $\gcd(n, k) = 1$  the Fourier  $n \times n$  matrix is shown to satisfy the conditions and error-correcting pairs are exhibited which then solves systems when  $A$  is a Fourier matrix. An algorithm for the Fourier  $n \times n$  matrix, with the proviso that  $\gcd(n, k) = 1$ , is then given in Section 8.1, Algorithm 8.2. When  $k = 1$  (that is, when the measurements are taken consecutively) the algorithm for the Fourier matrix case is similar to that obtained in [12]. This paper [12] also makes the point “to make the algorithm more robust to noise we have to increase the number of available samples .. and apply some denoising algorithm to the samples”.

The Vandermonde matrix  $A = V(x_1, x_2, \dots, x_n)$ , ( $x_i \neq 0$ ), in which the quotients  $x_i/x_j$ , for  $i \neq j$ , are not  $k^{\text{th}}$  roots of unity, where  $k$  is the arithmetic difference in  $J$ , is shown to satisfy the general requirements and error-correcting pairs are explicitly given for these cases. An algorithm for finding a solution of  $Aw = y$  for such a Vandermonde matrix  $A$  with rows  $E_i$  where  $\{E_j w | j \in J\}$  are known with  $J$  in arithmetic sequence with difference  $k$  such that  $x_i/x_j$ , ( $i \neq j, 1 \leq i, j \leq n$ ) is not a  $k^{\text{th}}$  root of unity), is given in Section 7, Algorithm 7.1.

The algorithms in general involve a maximum of  $O(nt)$  operations but in some cases a maximum of  $O(t^2)$  operations only is required. The Fourier  $n \times n$  matrix case requires  $\max(O(t^2), O(n \log n))$  operations and the operations are known to be particularly efficient and stable.

The technique involves considering the problem as a coding/decoding problem and then to find suitable decoding algorithms. A particularly useful decoding algorithm involves finding *error-correcting pairs* for the code. The method of error-correcting pairs is due jointly to Pelikaan [13] and Duursma & Kötter [4].

A technique is derived in Section 5 to deal with a type of random selection of rows of a matrix which has the property that the determinant of any square submatrix is non-zero. Matrices which satisfy the condition that the determinant of any square submatrix is non-zero include the Fourier  $n \times n$  matrices where  $n$  is prime (Chebotarëv’s Theorem), real Vandermonde matrices with positive (distinct) entries and Cauchy matrices.

## 2 Coding theory method

Consider  $w$  as the *error vector of a code*. As  $w$  has at most  $t$  non-zero entries, a  $t$ -error correcting code for which  $w$  is the error vector is then required. A method which can locate and identify the ‘errors’, which are then the entries of  $w$ , solves for  $w$ .

A basic reference for coding theory is [2]. A code  $C$  over a field  $F$  is a subset of  $F^n$  and all codes considered are linear. An  $(n, r)$  code is a code of length  $n$  and dimension  $r$  and an  $(n, r, d)$  code is a code of length  $n$ , dimension  $r$  and (minimum) distance  $d$ . An  $(n, r, \geq d)$  code is a code of length  $n$ , dimension  $r$  and distance  $\geq d$ . An mds (maximal distance separable) code is an  $(n, r)$  code of distance  $(n - r + 1)$ , that is, an mds code is an  $(n, r, n - r + 1)$  code.

### 2.1 Rows generating codes

Let the rows of an  $m \times n$  matrix  $A$  be denoted by  $\{E_0, E_1, \dots, E_{m-1}\}$  and assume these are linearly independent. Measurements are taken of  $Aw$ , that is certain  $E_j w$  are taken or known for  $j \in J = \{j_1, j_2, \dots, j_u\} \subset \{0, 1, \dots, m-1\}$  and it is given that  $u \geq 2t$  where  $t$  is the maximum number of non-zero entries of  $w$ . It is clear it may be assumed without loss of generality that  $m = n$  as just a subset of the rows of  $A$  are used.

Let  $C^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$ . Think of  $C^\perp$  as a code and its dual (orthogonal complement) is denoted by  $\mathcal{C}$ . Now  $C^\perp$  is an  $(n, u)$  code and so  $\mathcal{C}$  is an  $(n, n - u)$  code which has an  $(n - u) \times n$  generator matrix denoted by  $C$ . Thus  $v \in \mathcal{C}$  if and only if  $E_i v = 0$  for each  $E_i \in C^\perp$  or equivalently  $v \in \mathcal{C}$  if and only if  $\hat{C}v = 0_{u \times 1}$  where  $\hat{C}$  is the  $u \times n$  matrix with rows consisting of the elements  $\{E_{j_1}, E_{j_2}, \dots, E_{j_u}\}$ .

Then  $CC^{\hat{C}^T} = 0_{(n-u) \times u}$ , which is equivalent to  $\hat{C}C^T = 0_{u \times (n-u)}$ , is the set-up for the generator matrix/check matrix of a code and its dual.

If  $\mathcal{C}$  is a  $t$ -error correcting code then it may be used to obtain  $w$ , provided of course a practical decoding algorithm is available.

Now  $\mathcal{C}$  is an  $(n, n - u)$  code and is  $t$ -error correcting if its distance is  $\geq 2t + 1$ . The maximum distance

that  $\mathcal{C}$  can attain is  $u + 1$ . For  $u = 2t$  this requires  $\mathcal{C}$  to be an  $(n, n - 2t, 2t + 1)$  code, that is, it must be an mds code. Now  $\mathcal{C}$  is an mds  $(n, n - 2t, 2t + 1)$  code if and only if its dual  $\mathcal{C}^\perp$ , with matrix  $\hat{C}$ , is an (mds)  $(n, 2t, n - 2t + 1)$  code. The check matrix,  $\hat{C}$ , of  $\mathcal{C}$  is an  $(2t \times n)$  matrix. Thus  $\mathcal{C}$  has distance  $2t + 1$  if and only if any  $2t$  columns of  $\hat{C}$  are linearly independent – see for example Corollary 3.2.3 in [2] for details on this. Thus for  $u = 2t$  it is required that any  $2t$  columns of  $\hat{C}$ , which is a  $2t \times n$  matrix, are linearly independent.

For  $u > 2t$  it is required that  $\mathcal{C}$  be a  $(n, u, \geq (2t + 1))$  code. Now  $\hat{C}$ , an  $u \times n$  matrix, is the check matrix of  $\mathcal{C}$  and thus it is required that any  $2t$  columns of  $\hat{C}$  be linearly independent. If any  $u$  columns of  $\hat{C}$  are linearly independent then of course any  $2t$  columns are linearly independent and  $\mathcal{C}$  is at least  $t$ -error correcting.

There are a number of cases where it can be assured that  $\mathcal{C}$  is  $t$ -error correcting.

When  $A$  satisfies the property that the determinant of any square submatrix of  $A$  is non-zero then any choice of  $r$  rows of  $A$  gives an mds  $(n, r, n - r + 1)$  code, [8]. The Fourier  $n \times n$  matrix for a prime  $n$  has this property by a result of Chebotarëv<sup>1</sup>. Thus as shown in [8] any code obtained by taking  $(n - 2t)$  rows of this Fourier matrix gives an  $(n, n - 2t, 2t + 1)$  mds code. Hence when  $A$  is the Fourier  $n \times n$  matrix for  $n$  a prime any such  $\hat{C}$  has the required mds property. A Vandermonde real matrix with positive entries has this property, Corollary 6.5 below.

In general if  $V(x_1, x_2, \dots, x_n)$  is a Vandermonde matrix and the  $E_{j_k}$  in  $\mathcal{C}^\perp$  are evenly distributed with arithmetic difference  $k$  such that the ratios  $x_i/x_j$  for all  $i \neq j$  are not  $k^{th}$  roots of unity then  $\mathcal{C}^\perp$  is an mds codes, see Corollary 6.3 and Section 7 below. For a general  $n \times n$  Fourier matrix it will be shown in Section 8 that mds codes are obtained when the  $E_j$  in  $\mathcal{C}^\perp$  are evenly spaced with arithmetic difference  $k$  satisfying  $\gcd(n, k) = 1$ .

When  $A$  is a Cauchy matrix, it also has the property that the determinant of any submatrix is non-zero but this case can be highly unstable and a decoding method is not easy to obtain.

## 2.2 Unit-derived codes

Suppose  $AB = 1$  for  $n \times n$  matrices  $A, B$ . Then as shown in [9] taking any  $r$  rows of  $A$  gives a generator matrix of an  $(n, r)$  code and the check matrix may be obtained by deleting the corresponding  $r$  columns of  $B$ . Alternatively any  $r$  rows of  $A$  gives the check matrix of an  $(n, n - r)$  code whose generator matrix is obtained by deleting the corresponding  $r$  columns of  $B$ .

This is the situation we have for the underdetermined given system  $Aw = y$  when  $A$  is an  $n \times n$  matrix with inverse  $B$ .

## 2.3 Decode to solve

Suppose now that  $\hat{C}$  has the required property that any  $2t$  columns are linearly independent. Then  $\mathcal{C}$  has distance  $\geq 2t + 1$  and so the code can correct the ‘errors’; it can find the elements of  $w$  using the check matrix  $\hat{C}$ . The problem is to find a suitable decoding method, that is, a method to locate and quantify these errors. The method should be of reasonable complexity and stable for applications.

We show now that when the measurements are evenly spaced within certain matrices an error-correcting (decoding) method exists which identifies  $w$ . In general the identification can be done in at worst  $O(tn)$  operations but in some cases it may be done in at worst  $O(t^2)$  operations. In practical applications  $t$  is often much smaller than  $n$ .

## 2.4 Error-correcting pairs

The method of *error-correcting pairs*, when they can be shown to exist, may be used to locate and determine the ‘errors’, and these ‘errors’ then determine the elements of  $w$ . The method of error-locating and error-correcting pairs is due jointly to Pellikaan [13] and to Duursma and Kötter [4]. The

<sup>1</sup>A proof of this Chebotarëv theorem may be found in [5] and proofs also appear in the expository paper of P.Stevenhagen and H.W Lenstra [14]; paper [6] contains a relatively short proof. There are several other proofs in the literature some of which are referred to in [14]. Paper [15] contains a proof of Chebotarëv’s theorem and refers to it as ‘an uncertainty principle’.

method used here is based mainly on that of Pellikaan [13].

Let  $F$  be a field and  $\mathcal{C}$  a (linear) code over  $F$ . Write  $n(\mathcal{C})$  for the code length of  $\mathcal{C}$ , its minimum distance is denoted by  $d(\mathcal{C})$  and denote its dimension by  $k(\mathcal{C})$ .

Now  $w_i$  denotes the  $i^{\text{th}}$  component of  $w \in F^n$ . For any  $w \in F^n$  define the support of  $w$  by  $\text{supp}(w) = \{i | w_i \neq 0\}$  and the zero set of  $w$  by  $z(w) = \{i | w_i = 0\}$ . The weight of  $w$  is the number of non-zero coordinates of  $w$  and denote it by  $wt(w)$ . The number of elements of a set  $I$  is denoted by  $|I|$ . Thus  $wt(w) = |\text{supp}(w)|$ .

We say that  $w$  has  $t$  errors supported at  $I$  if  $w = c + e$  with  $c \in \mathcal{C}$  and  $I = \text{supp}(e)$  and  $|I| = t = d(w, \mathcal{C})$ . For  $\mathcal{C}$  a linear code, the vector space of  $F$  linear functionals on  $\mathcal{C}$  is denoted by  $\mathcal{C}^\vee$ .

The bilinear form  $\langle, \rangle$  is defined by  $\langle a, b \rangle = \sum_i a_i b_i$ . For a subset  $C$  of  $F^n$ , the dual  $C^\perp$  of  $C$  in  $F^n$  with respect to the bilinear form  $\langle, \rangle$  is defined by  $C^\perp = \{x | \langle x, c \rangle = 0, \forall c \in C\}$ .

The sum of two elements of  $F^n$  is defined by adding corresponding coordinates. Of use in these considerations is what is termed the *star multiplication*  $a * b$  of two elements  $a, b \in F^n$  defined by multiplying corresponding coordinates, that is  $(a * b)_i = a_i b_i$ . For subsets  $A$  and  $B$  of  $F^n$  denote the set  $\{a * b | a \in A, b \in B\}$  by  $A * B$ . If  $A$  is generated by  $X$  and  $B$  is generated by  $Y$  then  $A * B$  is generated by  $X * Y$ .

**Definition 2.1** Let  $\mathcal{C}$  be a linear code in  $F^n$ . Define the syndrome map of the code  $\mathcal{C}$  by  $s : F^n \rightarrow (\mathcal{C}^\perp)^\vee, w \mapsto (v \mapsto \langle v, w \rangle)$ .

For a received word  $w \in F^n$  we call  $s(w)$  the syndrome of  $w$  with respect to the code  $\mathcal{C}$ .

**Definition 2.2** Let  $A, B$  and  $C$  be linear codes in  $F^n$ . Define the error locator map  $E_w$  of a received word  $w$  with respect to the code  $C$  by  $E_w : A \rightarrow B^\vee, a \mapsto (b \mapsto \langle w, a * b \rangle)$ .

Remark: If  $A * B \subseteq C^\perp$  and  $w$  is a word with error  $e$ , then  $E_w = E_e$ .

**Definition 2.3** Suppose  $I = \{i_1, i_2, \dots, i_t\}$ , where  $1 \leq i_1 < \dots < i_t \leq n$ . Let  $A$  be a linear code in  $F^n$ . Define  $A(I) = \{a \in A | a_i = 0, \forall i \in I\}$ .

**Definition 2.4** Define the projection map  $\pi_I : F^n \rightarrow F^t$  by  $\pi_I(w) = (w_{i_1}, \dots, w_{i_t})$ . Define  $A_I = \pi_I(A)$ . Let  $e \in F^n$ . Denote  $\pi_I(e * A)$  by  $e A_I$ .

**Definition 2.5** Suppose  $I = \{i_1, i_2, \dots, i_s\}$ . Define the inclusion map  $i_I : F^t \rightarrow F^n$  by mapping the  $j^{\text{th}}$  component,  $w_j$  of  $w$  into the  $i_j^{\text{th}}$  coordinate for all  $j = 1, 2, \dots, t$  and zeros everywhere else.

Define the restricted syndrome map  $s_I : F^t \rightarrow (\mathcal{C}^\perp)^\vee$  by  $s_I = s * i_I$ .

**Definition 2.6** Let  $A, B$  and  $C$  be linear codes in  $F^n$ . We call  $(A, B)$  a  $t$ -error correcting pair for  $C$  if

- 1)  $A * B \subseteq C^\perp$
- 2)  $k(A) > t$
- 3)  $d(A) + d(C) > n$ ,
- 4)  $d(B^\perp) > t$ .

**Definition 2.7** For an element  $w \in F^n$  define  $E_w : A \rightarrow B^\vee, a \mapsto (b \mapsto \langle w, a * b \rangle)$ .

Now refer to the paper [13] and in particular Proposition 2.11 therein. The paper contains the following algorithm, Algorithm 2.3, for locating and determining the values of errors in the code  $C$  when error-correcting pairs exist for  $C$ :

**Algorithm 2.1** (see [13], Algorithm 2.3.):

- 1.1 Compute  $\ker(E_w)$ .
- 1.2 If  $\ker(E_w) = 0$ , then goto 3.2.
- 1.3 If  $\ker(E_w) \neq 0$ , then choose a nonzero element  $a \in \ker(E_w)$ .

Let  $J = z(a)$ .

2.1 Compute the space of solutions of  $s_J(x) = s(w)$ .

2.2 If  $s_J(x) = s(w)$  has no or more than one solution then goto 3.2.

2.3 If  $s_J(x) = s(w)$  has the unique solution  $x_0$ , then compute  $wt(x_0)$ .

2.4 If  $wt(x_0) > t$ , then goto 3.2.

3.1 Print: "The received word is decoded by"; Print:  $w - i_J(x_0)$ ; goto 4.

3.2 Print: "The received word has more than  $t$  errors."

4 End.

In our case the actual errors are the values required and so 3.1 is changed accordingly. Case 3.2 will not arise as by assumption  $w$  has at most  $t$  non-zero entries or else it will show up pointing out an error in this assumption.

### 3 Solve the system of equations by decoding

Recall the star product  $u * v$  of two vectors  $u, v \in F^n$ . This is defined by multiplying corresponding coordinates, that is  $(u * v)_i = u_i v_i$ . For subsets  $U$  and  $V$  of  $F^n$  denote the set  $\{u * v | u \in U, v \in V\}$  by  $U * V$ .

Consider now a matrix  $A$  with rows  $\{E_0, E_1, \dots, E_{n-1}\}$ . Assume the matrix  $A$  satisfies conditions (a) and (b) as follows:

(a)  $E_i * E_j = E_{i+j}$  for  $i + j \leq (n - 1)$ .

(b) Let  $J \subset \{0, 1, \dots, n - 1\}$  be in arithmetic sequence with  $|J| = r$ . Then the code generated by  $\{E_j, j \in J\}$  is an mds  $(n, r, n - r + 1)$  code.

In the above condition (a) it is required that  $i + j \leq (n - 1)$ . where  $A$  has rows  $\{E_0, E_1, \dots, E_{n-1}\}$ . When for example  $A$  is the Fourier  $n \times n$  matrix then  $E_{i+j}$  is always defined with  $E_{i+j} = E_{i+j \bmod n}$ . In other cases also  $E_i$  may be defined for all  $i \in \mathbb{Z}$  where  $E_i, 0 \leq i \leq (n - 1)$  correspond to the rows of  $A$  as for example when  $A$  is a Vandermonde matrix. In such cases the conditions (a) and (b) may be replaced as follows. Let  $A$  have rows  $\{E_0, E_1, \dots, E_{n-1}\}$  such that  $E_i$  are defined for  $i \geq 0$  (which coincide with rows of  $A$  for  $0 \leq i \leq n - 1$ ). Assume  $A$  satisfies conditions (A) and (B) as follows.

(A)  $E_i * E_j = E_{i+j}$ .

(B) Any  $J \subset \{0, 1, \dots, n - 1\}$  in arithmetic sequence is such that the code generated by  $\{E_j, j \in J\}$  is an mds  $(n, r, n - r + 1)$  code where  $|J| = r$ .

Only a subset of the rows of  $A$  are used in the general theory. We may assume  $A$  has first row  $E_0$  by the following consideration. Suppose  $A$  has rows numbered  $\{E_1, E_2, \dots, E_{n-1}\}$  satisfying conditions (a) and (b) or conditions (A) and (B) with  $1 \leq i, j$ . Introduce a first row  $E_0$  into  $A$  where  $E_0$  is the  $1 \times n$  vector consisting of all 1's; this new matrix will still be referred to as  $A$  and satisfies the required conditions with  $0 \leq i, j$ .

Assume then from now on in this section that the matrix  $A$  satisfies conditions (a) and (b) or where appropriate conditions (A) and (B). Matrices which satisfy conditions (a) and (b) or conditions (A) and (B) are given in subsequent sections.

Rows of  $A$  are given to form  $C^\perp = \{E_j | j \in J\}$ , as in Section 2.1, where now the  $E_j$  are evenly distributed, that is,  $C^\perp = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+(2t-1)j} \rangle$ . (It is assumed that  $2t < n$  and that  $E_k$  are defined for  $0 \leq k \leq i + (2t - 1)j$ .)

With this set-up it is possible to get a  $t$ -error correcting pair, (see definition 2.6), for  $\mathcal{C}$  the dual code of  $C^\perp$ . In these cases the vector  $w$  (from  $Aw$  where rows  $E_j w, j \in J$  are known) which has at most  $t$

non-zero entries, may be obtained by applying the method of Algorithm 2.1 above due to Pellikaan [13] to give an appropriate implementable algorithm in which to find  $w$ . It will be shown that the solution may be obtained in at most  $O(tn)$  operations and in some cases in at most  $O(t^2)$  operations.

Take initially the case  $C^\perp = \langle E_1, E_2, \dots, E_{2t} \rangle$ , that is, the  $E_i$  are consecutive starting at  $E_1$ ; the more general case will be dealt with similarly.

**Theorem 3.1** *Let  $C^\perp = \langle E_1, E_2, \dots, E_{2t} \rangle$  with  $2t \leq n$  and  $C$  is the dual of the code generated by  $C^\perp$ . Define  $U = \langle E_1, E_2, \dots, E_{t+1} \rangle, V = \langle E_0, E_1, \dots, E_{t-1} \rangle$ . Suppose that  $C^\perp, U, V$  generate mds codes. Then  $(U, V)$  is a  $t$ -error correcting pair for  $C$ .*

**Proof:** Now  $E_i * E_j = E_{i+j}$ . Then  $A * B \subseteq \langle E_1, E_2, \dots, E_{2t} \rangle \subseteq C^\perp$ .

Note that a code is an mds code if and only if its dual is an mds code.

Now  $C$  is an  $(n, n-2t, 2t+1)$  code,  $U$  is an  $(n, t+1, n-t)$  code,  $V$  is an  $(n, t, n-t+1)$  code and  $V^\perp$  is an  $(n, n-t, t+1)$  code. Thus  $k(U) = t+1 > t, d(U)+d(C) = (n-t)+(2t+1) = n+t+1 > n, d(V^\perp) = t+1 > t$  and so  $(U, V)$  is a  $t$ -error correcting pair for  $C$  (see Definition 2.6).  $\square$

In the general case we have the following. The proof is similar to the proof of Theorem 3.1 above. Let  $E_0$  be the vector with all  $1^s$  as entries. The suffices  $lj$  in the following theorems actually mean  $l * j$ , the multiplication of  $l$  by  $j$ .

**Theorem 3.2** *Let  $C^\perp = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+(2t-1)j} \rangle$ . The dual code of  $C^\perp$  is  $C$ . Define  $U = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+tj} \rangle, V = \langle E_0, E_{1j}, E_{2j}, \dots, E_{(t-1)j} \rangle$ .*

*Suppose  $C^\perp, U, V$  generate mds codes. Then  $(U, V)$  is a  $t$ -error correcting pair for  $C$ .*

In a set-up there may be more than one error-correcting pairs and it may be useful to consider others. For example we could interchange some of the elements of  $U, V$ .

**Theorem 3.3** *Let  $C^\perp = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+(2t-1)j} \rangle$ . The dual code of  $C^\perp$  is  $C$ . Suppose a vector  $E_{i-j}$  exists with  $E_{i-j} * E_j = E_i$ . Define  $U = \langle E_{i-j}, E_i, E_{i+j}, \dots, E_{i+(t-1)j} \rangle, V = \langle E_{1j}, E_{2j}, \dots, E_{tj} \rangle$ .*

*Suppose  $C^\perp, U, V$  generate mds codes. Then  $(U, V)$  is a  $t$ -error correcting pair for  $C$ .*

It is thus noted that there may exist a number of different error-correcting pairs for the same code.

**Example 3.1** *Let  $A$  have rows  $E_i$  with  $E_i * E_j = E_{i+j}$ . Denote  $E_i$  by  $i$  and thus  $E_i * E_j = E_{i+j}$  translates to  $i * j = i + j$ . Let  $C^\perp = \langle 5, 7, 9, 11, 13, 15 \rangle$  so that  $C$  is 3-error correcting (when  $C^\perp$  is mds). The following are 3-error-correcting pairs.*

- $U = \langle 5, 7, 9, 11 \rangle, V = \langle 0, 2, 4 \rangle$ .
- $U = \langle 3, 5, 7, 9 \rangle, V = \langle 2, 4, 6 \rangle$ .
- $U = \langle 1, 3, 5, 7 \rangle, V = \langle 4, 6, 8 \rangle$ .
- When  $-i$  exist (as for the Fourier matrix) it's clear that further error-correcting pairs for  $C$  can easily be found.

### 3.1 Interpretation

Consider  $C^\perp, U, V$  as in Theorem 3.2. Now apply Algorithm 2.1 (derived from [13]) using the error-correcting pairs found in Theorem 3.2. (Other correcting pairs, as shown can exist, may also be used.) We show that the error locations may be obtained from the matrix given in the following Theorem relative to the bases  $\{E_i, E_{i+j}, \dots, E_{i+tj}\}$  for  $U$  and  $\{\omega_0, \omega_1, \dots, \omega_{t-1}\}$  for  $V^\vee$ , where  $\omega_i : E_{kj} \mapsto \delta_{ik}$  for  $i = 0, 2, \dots, t-1$ . Write  $F_k = E_{i+(k-1)j}$  for  $k = 1, 2, \dots, 2t$ . Thus  $U$  has basis  $\{F_1, F_2, \dots, F_{t+1}\}$  and  $C^\perp$  has basis  $\{F_1, F_2, \dots, F_{2t}\}$ . Let  $\alpha_s = \langle w, F_s \rangle = F_s w = E_{i+(s-1)j} w$  for  $s = 1, \dots, 2t$  and these are known.

Recall, definition 2.7, that  $E_w : U \rightarrow V^\vee, u \mapsto (v \mapsto \langle w, u * v \rangle)$ .

**Theorem 3.4**  $E_w$  has the following matrix relative to the basis  $\{F_1, F_2, \dots, F_{t+1}\}$  for  $U$  and the basis  $\{\omega_1, \omega_2, \dots, \omega_t\}$  for  $V^\vee$ , where  $\omega_i : E_{kj} \mapsto \delta_{ik}$ .

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \alpha_3 & \alpha_4 & \dots & \alpha_{t+3} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$$

**Proof:**

Now  $E_w : U \rightarrow V^\vee, u \mapsto (v \mapsto \langle w, u * v \rangle)$ .

Thus  $E_w$  works as following on  $F_1$ :

$$F_1 \mapsto \begin{pmatrix} E_0 \mapsto \langle w, F_1 * E_0 \rangle \\ E_j \mapsto \langle w, F_1 * E_j \rangle \\ \vdots \\ E_{(t-1)j} \mapsto \langle w, F_1 * E_{(t-1)j} \rangle \end{pmatrix} = \begin{pmatrix} E_0 \mapsto \langle w, F_1 \rangle \\ E_j \mapsto \langle w, F_2 \rangle \\ \vdots \\ E_{(t-1)j} \mapsto \langle w, F_t \rangle \end{pmatrix} = \begin{pmatrix} E_0 \mapsto \alpha_1 \\ E_j \mapsto \alpha_2 \\ \vdots \\ E_{(t-1)j} \mapsto \alpha_t \end{pmatrix}.$$

Thus  $E_w : F_1 \mapsto \alpha_1 \omega_1 + \alpha_2 \omega_2 + \dots + \alpha_t \omega_t$ . Similarly  $E_w : F_i \mapsto \alpha_i \omega_1 + \alpha_{i+1} \omega_2 + \dots + \alpha_{i+t-1} \omega_t$ .

$$\text{Hence } E_w : (F_1, F_2, \dots, F_{t+1}) \mapsto (w_1, w_2, \dots, w_t) \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$$

Thus the matrix of  $E_w$  relative to bases  $\{F_1, F_2, \dots, F_{t+1}\}$  for  $U$  and the basis  $\{\omega_1, \omega_2, \dots, \omega_t\}$  for  $V^\vee$  is

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}.$$

□

The matrix in Theorem 3.4 is a Hankel matrix and its kernel in general can be obtained in at most  $O(t^2)$  operations. Just any non-zero element of the kernel is required.

It is then required to multiply a non-zero element of the kernel of the matrix by  $(F_1, F_2, \dots, F_{t+1})$  to get an actual kernel element of the mapping  $E_w$ .

Suppose then such an element  $a \in \ker E_w$  has been found. Let  $J = z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . It is now required to compute the space of solutions of  $s_J(x) = s(w)$ . Suppose  $J = \{i_1, i_2, \dots, i_t\}$  and let  $x \in F^n$ . Then  $s_J(x) = s * i_J(x)$ . Let  $i_J(x) = y$  and suppose now  $y = i_J(x)$  is  $x_1$  in  $i_1$  position,  $x_2$  in  $i_2$  position and in general  $x_k$  in  $i_k$  position and zeros elsewhere.

Now  $s : F^n \rightarrow (C^\perp)^\vee$  is  $u \mapsto (v \mapsto \langle v, u \rangle)$ . A basis for  $C^\perp$  is  $\{F_1, F_2, \dots, F_{2t}\}$ .

$$\text{Hence } s : w \mapsto \begin{pmatrix} F_1 \mapsto \langle F_1, w \rangle & = \alpha_1 \\ F_2 \mapsto \langle F_2, w \rangle & = \alpha_2 \\ \vdots & \vdots \\ F_{2t} \mapsto \langle F_{2t}, w \rangle & = \alpha_{2t} \end{pmatrix}.$$

Since  $F_i \in F^n$ , let  $F_i = (F_{i,1}, F_{i,2}, \dots, F_{i,n})$  for  $i = 1, 2, \dots, 2t$ .

Now

$s_J(x) = s * i_J(x)$  and so:

$$s_J : x \mapsto \begin{pmatrix} F_1 \mapsto \langle F_1, y \rangle & = & x_1 F_{1,j_1} + x_2 F_{1,j_2} + \dots + x_t F_{1,j_t} \\ F_2 \mapsto \langle F_2, y \rangle & = & x_1 F_{2,j_1} + x_2 F_{2,j_2} + \dots + x_t F_{2,j_t} \\ \vdots & \vdots & \vdots \\ F_{2t} \mapsto \langle F_{2t}, y \rangle & = & x_1 F_{2t,j_1} + x_2 F_{2t,j_2} + \dots + x_t F_{2t,j_t} \end{pmatrix}$$

Hence solving  $s_J(x) = s(w)$  reduces to solving the following:

$$\begin{pmatrix} F_{1,j_1} & F_{1,j_2} & \dots & F_{1,j_t} \\ F_{2,j_1} & F_{2,j_2} & \dots & F_{2,j_t} \\ \vdots & \vdots & \vdots & \vdots \\ F_{2t,j_1} & F_{2t,j_2} & \dots & F_{2t,j_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (1)$$

The value of  $w$  is then the solution of these equations with entries in appropriate places as determined by  $J$ . The values of  $F_{i,k}$  are known and in some cases have nice forms. The matrix in (1) can be of a special type (for example, submatrix of Vandermonde and/or consisting of roots of unity) enabling practical (easier) calculation of a solution to equations (1).

## 4 Algorithms

Now algorithms are given based on the results of Section 3 with which to solve the underdetermined systems in various cases. Suppose  $y = Ax$  where  $A$  is an  $n \times n$ ,  $w$  an  $n \times 1$  unknown vector and where  $u$  entries of  $y$  are known. It is given that  $w$  has at most  $t$  non-zero entries. Denote the rows of  $A$  by  $\{E_0, E_1, \dots, E_{n-1}\}$  and suppose that  $E_i * E_j = E_{i+j}$ .

Measurements  $E_j w$  (values of  $y$ ) are taken or known for  $j \in M = \{j_1, j_2, \dots, j_u\} \subset \{0, 1, \dots, (n-1)\}$  where  $u \geq 2t$ . Suppose the measurements satisfy the conditions of Theorem 3.2 with  $C^\perp = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+(2t-1)j} \rangle$  and  $C$  is the dual code of the code generated by  $C^\perp$ . We give an Algorithm to calculate the value of  $w$  under these conditions when the measurements are in an arithmetic progression (evenly distributed) and subject to conditions of Section 3.

### 4.1 Case $k = 1$

We first for clarity give the algorithm when  $M = \{1, 2, \dots, u\}$  and  $u = 2t$ . This is easier to explain and avoids the complicated notation necessary for the general case given below.

The set-up then is that  $A$  is an  $n \times n$  matrix with rows  $\{E_0, E_1, \dots, E_{n-1}\}$  and that measurements  $E_i w$  are taken for  $i = 1, 2, \dots, 2t$ . It is assumed that  $w$  has at most  $t$  non-zero entries. Then  $w$  is determined as follows: Let  $\alpha_i = \langle w, E_i \rangle = E_i w$  for  $i \in J = \{1, 2, \dots, 2t\}$ .

#### Algorithm 4.1

- Find a non-zero element  $x^T$  of the kernel of  $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$ .
- Let  $a = (E_1, E_2, \dots, E_{t+1})x^T$ . (Any non-zero multiple of  $a$  will suffice as we are only interested in the zero entries of  $a$ . Note that  $a$  is a  $1 \times n$  vector.)
- Let  $z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . Suppose  $z(a) = \{j_1, j_2, \dots, j_t\}$  and denote this set by  $J$ .
- Solve  $s_J(x) = s(w)$ . This reduces to solving the following. Here  $E_i = (E_{i,1}, E_{i,2}, \dots, E_{i,n})$ .

$$\begin{pmatrix} E_{1,j_1} & E_{1,j_2} & \dots & E_{1,j_t} \\ E_{2,j_1} & E_{2,j_2} & \dots & E_{2,j_t} \\ \vdots & \vdots & \vdots & \vdots \\ E_{2t,j_1} & E_{2t,j_2} & \dots & E_{2t,j_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (2)$$

- The value of  $w$  is then the solution of these equations with entries in appropriate places as determined by  $J$ .

The complexity of the operations is discussed in Section 4.2.



## 4.2 General case

Suppose  $A$  is an  $n \times n$  matrix with rows  $\{E_0, E_1, \dots, E_{n-1}\}$  satisfying  $E_i * E_j = E_{i+j}$ .<sup>2</sup>

Measurements  $E_j w$  are taken or known for  $j \in J = \{j_1, j_2, \dots, j_u\}$  where  $u \geq 2t$ . The elements in  $J$  are in arithmetic progression with difference  $k$  so that the satisfying  $\gcd(n, k) = 1$ . Then  $w$  is calculated by the following algorithm.

Let  $\alpha_k = \langle w, F_{j_k} \rangle = F_{j_k} w$  for  $j_k \in J$ . Define  $F_i = E_{j_i}$  for  $j_i \in J$  and  $F_0 = E_{j_1-k}$  with indices taken mod  $n$ . Let  $F_i = (F_{i,1}, F_{i,2}, \dots, F_{i,n})$ .

### Algorithm 4.2

- Find a non-zero element  $x^T$  of the kernel of  $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$ .
- Let  $a = (F_0, F_1, \dots, F_t)x^T$ . (Any non-zero multiple of  $a$  will suffice as we are only interested in the zero entries of  $a$ . Note that  $a$  is a  $1 \times n$  vector.)
- Let  $z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . Suppose  $z(a) = \{j_1, j_2, \dots, j_t\}$  and denote this set by  $J$ .
- Solve  $s_J(x) = s(w)$ . This reduces to solving the following:

$$\begin{pmatrix} F_{1,j_1} & F_{1,j_2} & \dots & F_{1,j_t} \\ F_{2,j_1} & F_{2,j_2} & \dots & F_{2,j_t} \\ \vdots & \vdots & \ddots & \vdots \\ F_{2t,j_1} & F_{2t,j_2} & \dots & F_{2t,j_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (3)$$

- The value of  $w$  is then the solution of these equations with entries in appropriate places as determined by  $J$ .

## 5 Random selection

This section initiates a method for working with randomly chosen error-correcting pairs. It is independent of subsequent sections.

Suppose the  $n \times n$  matrix  $A$  in the underdetermined system  $Aw = y$  has the property that the determinant of any square submatrix is non-zero. Then the choice of any  $r$  rows of  $A$  yields an mds  $(n, r, n - r + 1)$  code. Matrices which have this property are the Fourier  $n \times n$  matrices with  $n$  a prime (Chebotarëv's theorem), the Vandermonde real matrices with positive entries and Cauchy matrices.

When considering  $Aw = y$ , if any  $r$  rows of  $A$  are chosen for  $C^\perp$  (notation as in Section 2.1) then an mds code for  $C$  is obtained but we haven't an error-correcting pair to hand as when the rows are evenly distributed. Now approach the randomness from another point of view of *choose the error-correcting pair randomly* and this decides the rows to be chosen for the measurements (code); then the randomly chosen pair is an error-correcting pair for this code.

This section enables working with rows of matrices which have the property that the determinant of any square submatrix is non-zero as the 'samples' for  $Aw$ . However the systems in general may require more than  $2t$  samples when the  $w$  has just  $t$  non-zero entries.

Consider then the following Proposition of Duursma and Kötter [4].

(For  $U, V \in F^n$  let  $U * V$  denote the space generated by  $\{u * v | u \in U, v \in V\}$ .)

**Proposition 5.1** (See Proposition 1 of [4].) *Let  $U, V$  be mds codes with  $k(U) = t + 1, k(V) = t$ . Any code  $C \perp U * V$  has distance  $\geq 2t + 1$  and has  $t$ -error correcting pair  $(U, V)$ .*

<sup>2</sup>More generally it is sometimes enough that  $E_i * E_j = \alpha E_{i+j}$  for some scalar  $\alpha$  but this is not considered here.

### 5.0.1 Illustrative examples of random selection

The examples given necessarily have small length so they can be displayed but in general large length examples are easily obtained.

- Example 1: Let  $n = 19$  and let  $A$  be the  $19 \times 19$  Fourier matrix with rows  $\{E_0, E_1, \dots, E_{18}\}$ . As 19 is prime any choice of rows of  $A$  gives an mds code. We now manufacture a 3-error correcting code ( $t = 3$ ) with 3-error correcting pair. Then randomly choose 4 and 3 rows of  $A$ . Suppose then  $U = \langle E_1, E_3, E_6, E_{10} \rangle$ ,  $V = \langle E_0, E_5, E_8 \rangle$ . Then  $U * V = \langle E_1, E_3, E_6, E_8, E_9, E_{10}, E_{11}, E_{14}, E_{15}, E_{18} \rangle$  and let  $C^\perp = U * V$ . Then  $C$  is a code with distance  $\geq 2t + 1 = 7$ . Actually  $C^\perp$  is an  $(19, 10, 9)$  code and  $C$  is an  $(19, 9, 11)$  code. So in fact the code  $C$  is a 5-error correcting code but we just have a 3-error correcting pair.

- Example 2.

Now let  $A$  be as in Example 1. Here we produce a 5-error correcting pair by choosing randomly  $U = \langle E_1, E_3, E_6, E_{10}, E_{18} \rangle$  and then choosing  $V$  to be 4 of these say  $V = \langle E_1, E_3, E_6, E_{10} \rangle$ . Then let  $C^\perp = U * V$ . Now  $U * V$  has 13 elements and so  $C^\perp$  is a  $(19, 13, 6)$  code and  $C$  is a  $(19, 6, 14)$  code. Thus  $C$  is a 6 error correcting code and we have a 5 error correcting pair for it.

- Consider the Cauchy (which is Hilbert) matrix  $A = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & 1/5 & \dots \\ 1/3 & 1/4 & 1/5 & 1/6 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$

Denote the rows of  $A$  by  $E_1, E_2, \dots$ . Suppose we want a  $t$  error correcting code. Let  $U = \langle E_1, E_2, E_3 \rangle$ ,  $V = \langle E_1, E_2 \rangle$ . Then  $U * V = \langle E_1 * E_1, E_2 * E_2, E_3 * E_1, E_2 * E_2, E_2 * E_3 \rangle$  and let  $C^\perp = U * V$ .

Then

$$C^\perp = \langle (1, 1/4, 1/9, 1/16, \dots), (1/2, 1/6, 1/12, 1/20, \dots), (1/3, 1/8, 1/15, 1/24, \dots), (1/4, 1/9, 1/16, 1/25, \dots), (1/6, 1/12, 1/20, 1/30, \dots) \rangle$$

is the required code with which to take the ‘samples’. Now  $C$  is an  $(n, n - 5)$  code (provided the elements in  $C^\perp$  are independent) and is 2-error correcting with error locating pair  $(U, V)$ ; now  $C$  may be a  $(n, n - 5, 6)$  code but is by the theory a  $(n, n - 5, \geq 5)$  code. If the elements of  $C^\perp$  are not independent then  $C$  is an  $(n, n - 4, 5)$  code.

## 5.1 Method

Suppose now  $A$  is a matrix such that any square submatrix has non-zero determinant.

Now choose at random any  $t + 1$  rows of  $A$  to form  $U$  and then any  $t$  rows of  $A$  to form  $V$ . Then let  $C^\perp$  be the space generated by  $\{u * v | u \in U, v \in V\}$ . From this it is deduced that  $d(C) \geq 2t + 1$  and  $C$  has  $t$ -error correcting pair  $(U, V)$ . Then proceed as before in Section 3 to produce the decoding algorithm with the  $t$ -error correcting pair with which to solve  $Aw = y$  where  $w$  has at most  $t$  non-zero entries and  $E_j w$  are known for  $E_j \in C^\perp$ .

We don’t need the multiplicative property  $E_i * E_j = E_{i+j}$  on the rows of  $A$  although  $U * V$  could be large; the largest rank that  $U * V$  could have is  $t(t + 1)$  but can often be made of a smaller order. However selections can be made so that the resulting code has dimension of  $O(t)$ . This for example by choosing the rows in  $U$ ,  $|U| = t + 1$ , and in  $V$ ,  $|V| = t$  to be in arithmetic sequence with the same difference will give  $C^\perp = U * V$  with  $2t$  elements; variations of the differences will also give  $|U * V| = st$  for very small  $s$  (compared to  $t$ ).

Being able to randomly choose the error-correcting pairs and thus the measurements  $C^\perp$  suggests that encryption methods may possibly be introduced into the system.

Thus:

1. In  $Aw$  it is given that  $w$  has at most  $t$  non-zero entries and that the determinant of any square submatrix of  $A$  is non-zero.

2. Choose  $t + 1$  rows of  $A$  to form  $U$  and then  $t$  rows of  $A$  to form  $V$ .
3. Let  $C^\perp = U * V$ . Then  $C$  has distance  $\geq 2t + 1$  and  $(U, V)$  is a  $t$ -error correcting pair for  $C$ .
4. The measurements/samples  $E_j w$  are taken for  $E_j$  in a generating set of  $C^\perp$ .
5. The value of  $w$  is then determined by the decoding methods of Section 4. Details are omitted.

## 6 Determinants of Submatrices

The Vandermonde matrix  $V = V(x_1, x_2, \dots, x_n)$  is defined by

$$V = V(x_1, x_2, \dots, x_n) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{pmatrix}$$

It is well-known that the determinant of  $V$  is non-zero if and only if the  $x_i$  are distinct. Assume the  $x_i$  are non-zero.

**Proposition 6.1** *Let  $V = V(x_1, x_2, \dots, x_n)$  be a Vandermonde matrix with rows and columns numbered  $\{0, 1, \dots, n-1\}$ . Suppose rows  $\{i_1, i_2, \dots, i_s\}$  and columns  $\{j_1, j_2, \dots, j_s\}$  are chosen to form an  $s \times s$  submatrix  $S$  of  $V$  and that  $\{i_1, i_2, \dots, i_s\}$  are in arithmetic progression with arithmetic difference  $k$ . Then*

$$|S| = x_{k_1}^{i_1} x_{k_2}^{i_1} \dots x_{k_s}^{i_1} |V(x_{k_1}^k, x_{k_2}^k, \dots, x_{k_s}^k)|$$

**Proof:** Note that  $i_{l+1} - i_l = k$  for  $l = 1, 2, \dots, s-1$ , for  $k$  the fixed arithmetic difference.

$$\text{Now } S = \begin{pmatrix} x_{k_1}^{i_1} & x_{k_2}^{i_1} & \dots & x_{k_s}^{i_1} \\ x_{k_1}^{i_2} & x_{k_2}^{i_2} & \dots & x_{k_s}^{i_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k_1}^{i_s} & x_{k_2}^{i_s} & \dots & x_{k_s}^{i_s} \end{pmatrix} \text{ and so } |S| = \begin{vmatrix} x_{k_1}^{i_1} & x_{k_2}^{i_1} & \dots & x_{k_s}^{i_1} \\ x_{k_1}^{i_2} & x_{k_2}^{i_2} & \dots & x_{k_s}^{i_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k_1}^{i_s} & x_{k_2}^{i_s} & \dots & x_{k_s}^{i_s} \end{vmatrix}.$$

Hence by factoring out  $x_{k_i}$  from column  $i$  for  $i = 1, 2, \dots, s$  it follows that

$$|S| = x_{k_1}^{i_1} x_{k_2}^{i_1} \dots x_{k_s}^{i_1} \begin{vmatrix} 1 & 1 & \dots & 1 \\ x_{k_1}^k & x_{k_2}^k & \dots & x_{k_s}^k \\ x_{k_1}^{2k} & x_{k_2}^{2k} & \dots & x_{k_s}^{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k_1}^{(s-1)k} & x_{k_2}^{(s-1)k} & \dots & x_{k_s}^{(s-1)k} \end{vmatrix} = x_{k_1}^{i_1} x_{k_2}^{i_1} \dots x_{k_s}^{i_1} |V(x_{k_1}^k, x_{k_2}^k, \dots, x_{k_s}^k)|$$

□

A similar result holds when the columns  $\{j_1, j_2, \dots, j_s\}$  are in arithmetic progression.

**Corollary 6.1**  $|S| \neq 0$  if and only if  $|V(x_{k_1}^k, x_{k_2}^k, \dots, x_{k_s}^k)| \neq 0$ .

**Corollary 6.2**  $|S| \neq 0$  if and only if  $x_{k_i}^k \neq x_{k_j}^k$  for  $i \neq j, 1 \leq i, j \leq s$ . This happens if and only if  $(x_{k_i} x_{k_j}^{-1})^k \neq 1$  for  $i \neq j, 1 \leq i, j \leq s$ .

**Corollary 6.3**  $|S| \neq 0$  if and only if  $(x_{k_i} x_{k_j}^{-1})$  is not a  $k^{\text{th}}$  root of unity for  $i \neq j, 1 \leq i, j \leq s$ .

**Corollary 6.4** If the entries  $\{x_1, x_2, \dots, x_n\}$  are real then matrix  $|S| \neq 0$  if either (i)  $k$  is odd or (ii)  $k$  is even and  $x_i \neq -x_j$  for  $i \neq j$ .

**Corollary 6.5** If the entries  $\{x_1, x_2, \dots, x_n\}$  are real and positive then  $|S| \neq 0$ .

**Corollary 6.6** When  $x_i = \omega^{i-1}$  for a primitive  $n^{\text{th}}$  root of unity  $\omega$  (that is, when  $V$  is the Fourier  $n \times n$  matrix) and  $\gcd(k, n) = 1$  then  $|S| \neq 0$ .

**Proof:** If  $(x_{k_1}/x_{k_j})^k = 1$  then  $(\omega^{k_1-1}\omega^{1-k_j})^k = 1$  and so  $\omega^{k(k_1-k_j)} = 1$ . As  $\omega$  is a primitive  $n^{\text{th}}$  root of unity this implies that  $k(k_1 - k_j) \equiv 0 \pmod{n}$ . As  $\gcd(k, n) = 1$  this implies  $k_1 - k_j \equiv 0 \pmod{n}$  in which case  $k_1 = k_j$  as  $1 \leq k_1 < n, 1 \leq k_j < n$ . □

## 7 Vandermonde matrices

Let  $A = V(x_1, x_2, \dots, x_n)$  be a Vandermonde with rows  $\{E_0, E_1, \dots, E_{n-1}\}$ . Then  $E_i * E_j = E_{i+j}$ . As in Section 2.1 let  $C^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$ . By Corollary 6.3 if  $C^\perp$  has rows in arithmetic sequence with arithmetic difference  $k$  and the ratios  $x_i/x_j$  for  $i \neq j$  in  $A$  are not  $k^{th}$  roots of unity then  $C$  (the dual of  $C^\perp$ ) is an  $(n, n - 2t, 2t + 1)$  code and is  $t$ -error correcting with  $C^\perp$  as the check matrix. As shown in Section 3,  $C$  has an error correcting pair and Algorithm 4.2 in Section 4 may be applied.

Thus Vandermonde matrices for which  $x_i/x_j$  are not roots of unity are obvious choices in which to take rows of the matrix which are evenly spaced. Then Theorem 3.3 is satisfied and the decoding Algorithm 4.1 or 4.2 solves the underdetermined system  $Aw = y$  with Vandermonde matrix, provided the number of non-zero entries of  $w$  is limited.

Consider then a Vandermonde matrix

$$V = V(\beta_1, \beta_2, \dots, \beta_n) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \beta_1 & \beta_2 & \dots & \beta_n \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1^{n-1} & \beta_2^{n-1} & \dots & \beta_n^{n-1} \end{pmatrix}$$

We assume the  $\beta_i$  are distinct and non-zero.

Define  $E_k$  to be  $(\beta_1^k, \beta_2^k, \dots, \beta_n^k)$  for any  $k \in \mathbb{Z}$ . The rows of  $V$  are  $\{E_0, E_1, \dots, E_{n-1}\}$ .

**Lemma 7.1**  $E_i * E_j = E_{i+j}$ .

Thus we obtain the following set-up. Let  $A = V(\beta_1, \beta_2, \dots, \beta_n)$  and  $Aw = y$ . Measurements  $E_j w$  (values of  $y$ ) are taken or known for  $j \in M = \{j_1, j_2, \dots, j_u\} \subset \{0, 1, \dots, n-1\}$  where  $u \geq 2t$ . The elements in  $M$  are in arithmetic progression with difference  $k$  and  $\beta_i/\beta_j$  is not a  $k^{th}$  root of unity for  $i \neq j$ .

The following Algorithm 7.1 finds  $w$ ; this is special case of Algorithm 4.2 but can be read here independently of this.

Define  $F_i = E_{j_i}$  for  $j_i \in J$ . Let  $\alpha_i = \langle w, E_{j_i} \rangle = E_{j_i} w$  for  $j_i \in J$ . Let  $F_i = E_{j_i}$  for  $j_i \in J$ . Thus  $\alpha_i = \langle w, F_i \rangle$ .

**Algorithm 7.1**

(i) Find a non-zero element  $v^T$  of the kernel of  $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$ .

(ii) Let  $a = (F_1, F_2, \dots, F_{t+1})v^T$ .

(iii) Suppose  $v^T = (v_1, v_2, \dots, v_{t+1})^T$ . The  $i^{th}$  component of  $a$  is  $(v_1\beta_i^{j_1} + v_2\beta_i^{j_1+k} + \dots + v_{t+1}\beta_i^{j_1+tk})$ ; we are interested in when this is 0.

The  $i^{th}$  component of  $a$  is 0 if and only if  $v_1 + v_2\beta_i^k + v_3\beta_i^{2k} + \dots + v_{t+1}\beta_i^{tk} = 0$ .

(iv) Let  $z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . Suppose  $z(a) = \{i_1, i_2, \dots, i_t\}$  and denote this set by  $J$ .

(v) Solve  $s_J(x) = s(w)$ . This reduces to solving the following:

$$\begin{pmatrix} \beta_{i_1}^{j_1} & \beta_{i_2}^{j_1} & \dots & \beta_{i_t}^{j_1} \\ \beta_{i_1}^{j_2} & \beta_{i_2}^{j_2} & \dots & \beta_{i_t}^{j_2} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{i_1}^{j_{2t}} & \beta_{i_2}^{j_{2t}} & \dots & \beta_{i_t}^{j_{2t}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (4)$$

Since the elements in  $M$  have arithmetic difference  $k$  so that  $j_s = i_1 + (s-1)k$  for  $1 \leq s \leq 2t$ , this equation (4) is equivalent to

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \beta_{i_1}^k & \beta_{i_2}^k & \dots & \beta_{i_t}^k \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{i_1}^{(2t-1)k} & \beta_{i_2}^{(2t-1)k} & \dots & \beta_{i_t}^{(2t-1)k} \end{pmatrix} \begin{pmatrix} \beta_{i_1}^{j_1} x_1 \\ \beta_{i_2}^{j_1} x_2 \\ \vdots \\ \beta_{i_t}^{j_1} x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (5)$$

(vi) Then  $x = (x_1, x_2, \dots, x_t)$  is obtained from these equations (5) (or from (4)) and  $w$  has entries  $x_i$  in positions as determined by  $J$  and zeros elsewhere.

The matrix in (5) is a Vandermonde matrix. It is sufficient to solve the first  $t$  equations and the inverse of such a  $t \times t$  Vandermonde type matrix may be obtained in  $O(t^2)$  operations. In connection with item (i), finding a non-zero element of the kernel of a Hankel  $t \times (t+1)$  matrix can be done in  $O(t^2)$  operations.

In connection with item (iii), consider  $f(x) = v_1 + v_2x + v_3x^2 + \dots + v_{t+1}x^t$ . It is required to find those  $\beta_i$  for which  $f(\beta_i^k) = 0$ . By Horner's method  $f(\beta_i^k)$  may be determined in  $O(t)$  operations and thus finding all  $i$  for which  $f(\beta_i^k) = 0$  can be done in  $O(nt)$  operations. Choose  $j \in J$  for item (iv) if  $f(\beta_j^k) = 0$ . Finding the zeros of  $f(x)$  takes the maximum of  $O(nt)$  operations and all other operations take a maximum of  $O(t^2)$  operations.

Calculations with Vandermonde type matrices obtained from the Fourier matrix are known to be stable.

### 7.0.1 Which are best?

A question then is which Vandermonde matrices are best for working with Algorithm 7.1. The Fourier matrix cases, which have entries in  $\mathbb{C}$ , are dealt with in Section 8.

Which Vandermonde real matrices are best?

Possibilities for investigation include

$$V = V(\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n}) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ (\frac{1}{2})^{n-1} & (\frac{1}{3})^{n-1} & \dots & (\frac{1}{n})^{n-1} \end{pmatrix} \text{ and}$$

$$V = V(\alpha, \alpha^2, \dots, \alpha^n) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha & \alpha^2 & \dots & \alpha^n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{n-1} & \alpha^{2(n-1)} & \dots & \alpha^{n(n-1)} \end{pmatrix} \text{ where } \alpha^i \alpha^{-j} \text{ is not a } k^{\text{th}} \text{ root of unity for}$$

$i \neq j, 1 \leq i, j \leq n$ .

## 8 Fourier matrix

Suppose now that  $A$  is the Fourier  $n \times n$  matrix with rows  $\{E_0, E_1, \dots, E_{n-1}\}$ . Measurements are taken of  $Aw$ , that is certain  $E_j w$  are taken or known for  $j \in J = \{j_1, j_2, \dots, j_u\}$  and it is given that  $u \geq 2t$  where  $t$  is the maximum number of non-zero entries of  $w$ .

**Theorem 8.1** Suppose the  $E_j$  in  $C^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$  are evenly spaced with arithmetic difference  $k$  satisfying  $\gcd(n, k) = 1$ . Then any  $u \times u$  square submatrix of  $\hat{C}$  has non-zero determinant.

**Proof:**

This follows directly from Corollary 6.6. □

Proposition 7 of [11] may also be used to prove Theorem 8.1 above. This Proposition 7 of [11] is analogous to Chebotarëv's theorem.

**Corollary 8.1** *Let  $\mathcal{C}$  be the code with check matrix from  $C^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$  where the  $E_{i_j}$  are evenly spaced with arithmetic difference  $k$  satisfying  $\gcd(n, k) = 1$ . Then  $\mathcal{C}$  is an mds  $(n, n - u, u + 1)$  code.*

Consider cases where  $u > 2t$ . Here  $\hat{C}$  is a  $(n, u)$  matrix and  $C$  is a  $(n, n - u)$  matrix. It is required that  $\mathcal{C}$  be a  $t$ -error correcting code and thus it is required that  $C$  be a  $(n, n - u, \geq (2t + 1))$  code. For this to happen it is required that any  $2t$  columns of  $\hat{C}$  be linearly independent.

Let  $A$  be Fourier  $n \times n$  matrix with rows  $\{E_0, E_1, \dots, E_{n-1}\}$ . When  $n$  is prime the code generated by  $\langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$  is an  $(n, u, n - u + 1)$  code; see [8]. In this case then  $C^\perp = \langle E_{j_1}, E_{j_2}, \dots, E_{j_u} \rangle$  with  $u = 2t$  generates an  $(n, 2t, n - 2t + 1)$  code and  $\mathcal{C}$ , its dual, is an  $(n, n - 2t, 2t + 1)$  code. Thus  $\mathcal{C}$  is a  $t$ -error correcting code. Now it is required to find a decoding algorithm for  $w$  as an error word of this code.

Assume that the  $E_{j_k}$  are evenly distributed, that is,  $C^\perp = \langle E_i, E_{i+j}, E_{i+2j}, \dots, E_{i+(2t-1)j} \rangle$  where suffices are taken mod  $n$ .

## 8.1 Algorithm for Fourier

This Algorithm is a special case of previous algorithms but can be read here independently.

Suppose  $y = Ax$  where  $A$  is an  $n \times n$  Fourier matrix,  $w$  an  $n \times 1$  unknown vector and where  $u$  entries of  $y$  are known. It is given that  $w$  has at most  $t$  non-zero entries and that  $u \geq 2t$ . Denote the rows of  $A$  by  $\{E_0, E_1, \dots, E_{n-1}\}$ .

Measurements  $E_j w$  (values of  $y$ ) are taken or known for  $j \in J = \{j_1, j_2, \dots, j_u\}$  where  $u \geq 2t$ . We give an Algorithm to calculate the value of  $w$  when the measurements are in an arithmetic progression (evenly distributed) with difference  $k$  satisfying  $\gcd(n, k) = 1$ .

### 8.1.1 Case $k = 1$

We first for clarity give the algorithm when  $K = \{1, 2, \dots, 2t\}$ . This is easier to explain and avoids the complicated notation necessary for the general case given below. The results and algorithm obtained in this case, where the measurements are taken consecutively, are similar to those in [12].

The set-up then is that  $A$  is the Fourier  $n \times n$  matrix with rows  $\{E_0, E_1, \dots, E_{n-1}\}$  and that measurements  $E_i w$  are taken for  $i = 1, 2, \dots, 2t$ . It is assumed that  $w$  has at most  $t$  non-zero entries. Then  $w$  is determined as follows: Let  $\alpha_i = \langle w, E_i \rangle = E_i w$  for  $i \in J = \{1, 2, \dots, 2t\}$ .

### Algorithm 8.1

1. Find a non-zero element  $x^T$  of the kernel of  $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$ .
2. Let  $a = (E_1, E_2, \dots, E_{t+1})x^T$ . (Any non-zero multiple of  $a$  will suffice as we are only interested in the zero entries of  $a$ . Note that  $a$  is a  $1 \times n$  vector.)
3. Let  $z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . Suppose  $z(a) = \{i_1, i_2, \dots, i_t\}$  and denote this set by  $J$ .
4. Solve  $s_J(x) = s(w)$ . This reduces to solving the following:

$$\begin{pmatrix} \omega^{i_1-1} & \omega^{i_2-1} & \dots & \omega^{i_t-1} \\ \omega^{2(i_1-1)} & \omega^{2(i_2-1)} & \dots & \omega^{2(i_t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{2t(i_1-1)} & \omega^{2t(i_2-1)} & \dots & \omega^{2t(i_t-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (6)$$

This is equivalent to solving the following:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \omega^{2(i_1-1)} & \omega^{2(i_2-1)} & \dots & \omega^{2(i_t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{2t(i_1-1)} & \omega^{2t(i_2-1)} & \dots & \omega^{2t(i_t-1)} \end{pmatrix} \begin{pmatrix} \omega^{i_1-1}x_1 \\ \omega^{i_2-1}x_2 \\ \vdots \\ \omega^{i_t-1}x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (7)$$

5. The value of  $w$  is then obtained from the solution  $x = (x_1, x_2, \dots, x_t)$  of equations (7) (or equations (6)) with entries  $x_i$  in positions as determined by  $J$  and zero elsewhere.

The complexity of the operations is discussed in Section 8.1.3.

### 8.1.2 General Fourier case

Suppose  $A$  is an  $n \times n$  Fourier matrix. Denote the rows of  $A$  by  $\{E_0, E_1, \dots, E_{n-1}\}$ .

Measurements  $E_j w$  are taken or known for  $j \in M = \{j_1, j_2, \dots, j_{2t}\} \subset \{0, 1, \dots, n-1\}$ . The elements in  $M$  are in arithmetic progression with difference  $k$  satisfying  $\gcd(n, k) = 1$ . Thus  $j_s = j_1 + (s-1)k$  for  $s = 1, 2, \dots, 2t$ . Then  $w$  is calculated by the following algorithm.

Let  $\alpha_k = \langle w, E_{j_k} \rangle = E_{j_k} w$  for  $j_k \in J$ . Define  $F_i = E_{j_i}$  for  $j_i \in J$ . Thus  $\alpha_k = \langle w, F_k \rangle = F_k w$ .

#### Algorithm 8.2

- (i) Find a non-zero element  $v^T$  of the kernel of  $E = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{t+1} \\ \alpha_2 & \alpha_3 & \dots & \alpha_{t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_t & \alpha_{t+1} & \dots & \alpha_{2t} \end{pmatrix}$ .
- (ii) Let  $a = (F_1, F_2, \dots, F_{t+1})v^T$ .
- (iii) Suppose  $v^T = (v_1, v_2, \dots, v_{t+1})^T$ . The  $i^{th}$  component of  $a$  is  $v_1 \omega^{(j_1)(i-1)} + v_2 \omega^{(j_1+k)(i-1)} + \dots + v_{t+1} \omega^{(j_1+tk)(i-1)}$ ; we are interested in when this is 0. The  $i^{th}$  component of  $a$  is 0 if and only if  $v_1 + v_2 \omega^{(i-1)k} + v_3 \omega^{(i-1)2k} + \dots + v_{t+1} \omega^{(i-1)tk} = 0$ .
- (iv) Let  $z(a) = \{j | a_j = 0\}$  which is the set of locations of the zero coordinates of  $a$ . Suppose  $z(a) = \{i_1, i_2, \dots, i_t\}$  and denote this set by  $J$ .
- (v) Solve  $s_J(x) = s(w)$ . This reduces to solving the following:

$$\begin{pmatrix} \omega^{j_1(i_1-1)} & \omega^{j_1(i_2-1)} & \dots & \omega^{j_1(i_t-1)} \\ \omega^{j_2(i_1-1)} & \omega^{j_2(i_2-1)} & \dots & \omega^{j_2(i_t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{j_{2t}(i_1-1)} & \omega^{j_{2t}(i_2-1)} & \dots & \omega^{j_{2t}(i_t-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (8)$$

Since  $j_s = j_1 + k(s-1)$  this reduces to solving the following system of equations:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \omega^{k(i_1-1)} & \omega^{k(i_2-1)} & \dots & \omega^{k(i_t-1)} \\ \omega^{2k(i_1-1)} & \omega^{2k(i_2-1)} & \dots & \omega^{2k(i_t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{(2t-1)k(i_1-1)} & \omega^{(2t-1)k(i_2-1)} & \dots & \omega^{(2t-1)k(i_t-1)} \end{pmatrix} \begin{pmatrix} \omega^{j_1(i_1-1)}x_1 \\ \omega^{j_1(i_2-1)}x_2 \\ \vdots \\ \omega^{j_1(i_t-1)}x_t \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2t} \end{pmatrix} \quad (9)$$

- (vi) Then  $x = (x_1, x_2, \dots, x_t)$  is obtained from these equations (9) from which  $w$  is derived with entries  $x_i$  in positions as determined by  $J$ .

### 8.1.3 Complexity

Finding the kernel of  $E$  is of  $O(t^2)$  as it involves finding the kernel of a  $t \times (t + 1)$  Hankel matrix, which in this case has dimension 1 in order to satisfy the given conditions. Superfast algorithms of  $O(t \log^2 t)$  with which to find the kernel of a Hankel matrix have been proposed. Item (iii), as already pointed out in Section 7, can be done in  $O(tn)$  operations; however by considering a Fourier Transform of  $(v_1, v_2, \dots, v_t, 0, \dots, 0)$  it can be performed in  $O(n \log n)$  operations by a Fast Fourier Transform.

The matrix (8) is a special Vandermonde type involving roots of unity only. There is a formula for the inverse of any Vandermonde matrix, the Björk, Pereyra method [1], which involves  $O(t^2)$  operations. Finding the inverse of a Vandermonde matrix with roots of unity is known to be particularly stable. The method of Björk, Pereyra [1] involves divisions by  $(\alpha_i - \alpha_j)$  where  $\alpha_i \neq \alpha_j$  and in these cases the  $\alpha_k$  are roots of unity. The system (8) or (9) could also be solved using the Forney Algorithm/formula, see [2] Chapter 7.

## References

- [1] A. Björck and V. Pereyra, “Solution of Vandermonde Systems of Equations”, *Mathematics of Computation*, 24, no. 112, 893-903, 1970.
- [2] Richard E. Blahut, *Algebraic Codes for data transmission*, Cambridge University Press, 2003.
- [3] E. J. Candès, J. K. Romberg, T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete Fourier information”, *IEEE Trans. Inf. Theory* 52 (8), 4895-909, 2006.
- [4] I. Duursma, R. Kötter, “Error-locating pairs for Cyclic Codes”, *IEEE Trans. in Inf. Theory*, 40, 1108-1121, 1994.
- [5] R. J. Evans and I. M. Isaacs, “Generalized Vandermonde determinants and roots of unity of prime order”, *Proc. of Amer. Math. Soc.* 58, 51-54, 1977.
- [6] P. E. Frenkel, “Simple proof of Chebotarëv’s theorem on roots of unity”, arXiv:math/0312398.
- [7] A. Hormati and M. Vetterli, “Annihilating filter-based decoding in the compressed sensing framework”, *Proc. SPIE, Wavelets XII*, vol. 6701, 1-10, 2007.
- [8] Barry Hurley and Ted Hurley, “Systems of MDS codes from units and idempotents”, *Discrete Mathematics*, 335, 81-91 2014.
- [9] Paul Hurley and Ted Hurley, “Codes from zero-divisors and units in group rings”, *Int. J. Inform. and Coding Theory*, 1, 57-87, 2009.
- [10] Paul Hurley and Ted Hurley, “Block codes from matrix and group rings”, Chapter 5, 159-194, in *Selected Topics in Information and Coding Theory* eds. I. Woungang, S. Misra, S.C. Misra, World Scientific 2010.
- [11] M. Ram Murty, Junho Peter Whang, “The uncertainty principle and a generalization of a theorem of Tao”, *Linear Alg. Applies.*, 437, 214-220, 2012.
- [12] Jon Oñativia, Yue Lu, Pier Dragotti, “Finite Dimensional FRI”, *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 1808-1812, 2014.
- [13] Ruud Pellikaan, “On decoding by error location and dependent sets of error positions”, *Discrete Mathematics*, 106/107, 369-381, 1992.
- [14] P. Stevenhagen and H. W. Lenstra, Jr., “Chebotarëv and his density theorem”, *Math. Intell.* 18, 26-37, 1996.
- [15] Terence Tao, “An uncertainty principle for groups of prime power order”, *Math. Res. Lett.*, 12, no. 1, 121-127, 2005.